Amendment to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of forwarding a network packet comprising~~es~~:

reading a table containing a plurality of flags to determine which of the plurality of flags is set or cleared;

determining if the table returned ~~a~~ decapsulate and ~~or~~ encapsulate flags;

adding a decapsulate byte count to a packet start offset and subtracting an encapsulate byte count from the packet start offset if the decapsulate and encapsulate flags are set; and

prepending encapsulate bytes to the packet.

2. (Previously Presented) The method of claim 1 wherein the table is populated with forwarding information.

3. (Previously Presented) The method of claim 1 wherein a forwarding table structure includes a control and management structure including a network stack interface and table managers.

4. (Currently Amended) The method of claim 3 wherein the table managers manage routing tables and can include a plurality of tables including a layer 3 ~~4~~ connection table, a layer 3

destination table, a layer 2 bridge table and a layer 2 connection table.

5. (Previously Presented) The method of claim 1 wherein the table includes a flag to indicate~~s~~ whether the bytes should be stripped from the packet and a field that indicates the number of bytes to be stripped.

6. (Previously Presented) The method of claim 1 wherein the table includes a field that specifies decapsulation of header layers up to a specified layer.

7. (Previously Presented) The method of claim 1 wherein the table includes a field that specifies an identifier of the current packet encapsulation type.

8. (Previously Presented) The method of claim 1 wherein the table includes a flag that indicates whether bytes should be prepended to the packet, a field that specifies the number of bytes and the bytes to be encapsulated.

9. (Previously Presented) The method of claim 1 wherein the table includes a Next Table Type field which indicates that a further lookup is required and identifies the table type.

10. (Canceled).

11. (Currently Amended) A method for encapsulating / decapsulating packets comprising:

receiving a packet;

reading in a first header of the packet and performing a layer 2 look-up reading a connection table which returns parameters;

determining if the table returned ~~a~~ decapsulate and ~~or~~ encapsulate flags;

adding a decapsulate byte count to a packet start offset and subtracting an encapsulate byte count from the packet start offset if the decapsulate and encapsulate flags are set; and

prepending encapsulate bytes to the packet.

12. (Previously Presented) The method of claim 11 further comprising:

determining if there is a next table to examine by looking at the blank field in the currently read table.

13. (Original) The method of claim 12 wherein if there is a next table,

parsing the next header and fetch and read the next table.

14. (Currently Amended) The method of claim 11 wherein if the decapsulate and encapsulate flags were not set,

determine if the encapsulate flag or the decapsulate flag were set.

15. (Currently Amended) The method of claim 11 wherein if the encapsulate flag was set,

subtract the encapsulate flag byte count from the start offset and prepend the encapsulate flags to the packet.

16. (Currently Amended) The method of claim 11 wherein if the decapsualte flag was set add a decapsulate byte count to the buffer offset and check the next table.

17. (Original) The method of claim 11 wherein the packet is comprised of one or more headers followed by a payload, the method further comprises:

copying the payload portion of the packet to a packet buffer.

18. (Original) The method of claim 17 wherein copying may place the packet at an offset in the buffer to make room for any new header that could be prepended to the packet for packet forwarding.

19. (Currently Amended) A computer program product residing on a computer readable media for forwarding a network packet comprises instructions to cause a computer to:

read a table containing a plurality of flags to determine, which of the plurality of flags is set or cleared;

determine if the table returned ~~a~~ decapsulate and ~~or~~ encapsulate flags;

add a decapsulate byte count to a packet start offset and subtract an encapsulate byte count from the packet start offset if the decapsulate and encapsulate flags are set; and

prepend encapsulate bytes to the packet.

20. (Previously Presented) The computer program product of claim 19 wherein the table is populated with forwarding information.

21. (Previously Presented) The computer program product of claim 19 wherein a forwarding table structures include a control and management structure including a network stack interface and table managers.

22. (Canceled).

23. (Currently Amended) A computer program product residing on a computer readable media for forwarding a network packet comprises instructions to cause a computer to:

receive a packet;

read in a first header of the packet and perform a layer 2 look-up reading a connection table which return parameters;

determine if the table returned ~~a~~ decapsulate and ~~or~~ encapsulate flags;

add a decapsulate byte count to a packet start offset and subtracting an encapsulate byte count from the packet start offset if the decapsulate and encapsulate flags are set; and

prepend encapsulate bytes to the packet.

24. (Previously Presented) The computer program product of claim 23 further comprising instructions to:

determine if there is a next table to examine by looking at a blank field in the currently read table.

25. (Original) The computer program product of claim 24 wherein if there is a next table, the computer program executes instructions to:

parse the next header and fetch and read the next table.

26. (Previously Presented) The computer program product of claim 23 wherein the packet is comprised of one or more headers followed by a payload, the computer program product further executes instructions to:

copy the payload portion of the packet to a packet buffer.

27. (Previously Presented) The computer program product of claim 26 wherein instructions to copy place the packet at an offset in the buffer to make room for any new header that could be prepended to the packet for packet forwarding.

28. (Currently Amended) A processor for processing a network packet comprises:

a computer storage media storing instructions to cause a computer to:

read a table containing a plurality of flags to determine, which of the plurality of flags is set or cleared;

determine if the table returned ~~a~~ decapsulate and ~~or~~ encapsulate flags;

add a decapsulate byte count to a packet start offset and subtract an encapsulate byte count from the packet start offset if the decapsulate and encapsulate flags are set; and

prepend the encapsulate bytes to the packet.

29. (Original) The processor of claim 28 wherein the table contains forwarding information.

30-31. (Canceled).

32. (Currently Amended) An apparatus comprising:

means for reading a table containing a plurality of flags to determine which of the plurality of flags is set or cleared;

means for determining if the table returned ~~a~~ decapsulate and ~~or~~ encapsulate flags;

means for adding a decapsulate byte count to a packet start offset and for subtracting an encapsulate byte count from the packet start offset if the decapsulate and encapsulate flags are set; and

means for prepending the encapsulate bytes to the packet.

33. (Previously Presented) The apparatus of claim 32 wherein the table is populated with forwarding information.

34. (Previously Presented) The apparatus of claim 32 wherein a forwarding table structure includes a control and management structure including a network stack interface and table managers.

35. (Currently Amended) The apparatus of claim 34 wherein the table managers manage routing tables and can include a plurality of tables including a layer 3 4 connection table, a layer 3 destination table, a layer 2 bridge table and a layer 2 connection table.